CLUSTER HPC LOVELACE
Guía básica de utilización v.1.0 – 20/01/2016

GUÍA BÁSICA DE USO DEL CLUSTER HPC LOVELACE

Este documento tiene como objetivo proporcionar los datos y aspectos básicos para la conexión y utilización del cluster HPC LOVELACE.

Índice:

DESCRIPCIÓN DEL CLUSTER

Se trata de un sistema compuesto por:

- Un nodo maestro [master] desde a donde se hace login, y que permite la submisión de trabajos y hace las labores de control, monitorización y gestión de recursos.
  - FUJITSU PRIMERGY CX2550 M1/D3343-A1
  - 2 x Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz con 6 cores cada uno [12 cores]
  - 64 GB de memoria RAM
- 20 nodos de cálculo [node01-20] con las siguientes características:
  - FUJITSU PRIMERGY CX2550 M1/D3343-A1
  - 2 CPUS Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz (8 cores, 20 Mb de caché) [16 cores]
  - 32 GB de memoria RAM , a 1,8 GHz
- 3 nodos de cálculo [node101-103] con las siguientes características
  - FUJITSU PRIMERGY CX2550 M1/D3343-A1
  - 2 CPUS Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz (8 cores, 20 Mb de caché)
  - 512 GB de memoria RAM
- 1 nodo de cálculo [node21] con tarjetas Xeon Phi
  - FUJITSU PRIMERGY CX2570 M1
  - 2 CPUS Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz (8 cores, 20 Mb de caché)
  - 256 GB de memoria RAM, a 1,8GHz
  - 2 x Intel Corporation Xeon Phi coprocessor 5100 series (rev 11)
- 1 nodo de cálculo [node22] con tarjetas NVIDIA Tesla
  - FUJITSU PRIMERGY CX2570 M1
  - 2 CPUS Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz (8 cores, 20 Mb de caché)
  - 256 GB de memoria RAM, a 1,8GHz
  - 2 GPU NVIDIA Corporation GK110GL [Tesla K20m] (rev a1)

SISTEMA DE FICHEROS Y POLÍTICAS DE USO

VARIABLE DE ENTORNO	RUTA	PROPÓSITO	CUOTA	POLÍTICA DE USO
$HOME	/LUSTRE/users/<cuenta>	Almacenamiento centralizado y compartido entre todos los nodos del cluster. Espacio de lectura y escritura global de usuario.	250 GB	Espacio disponible para la cuenta de usuario con cualquier propósito. No se elimina pero no se hace copia de seguridad automática.

$LOC_SCRATCH	/scratch/<cuenta>	Almacenamiento local a cada nodo para ficheros temporales en procesos paralelos	1 GB	Se eliminan los ficheros más antiguos de 10 días.
$SCRATCHDIR **$TMPDIR** **$TMP**	/LUSTRE/scratch	Almacenamiento global para todos los nodos para ficheros temporales		Se eliminan los ficheros más antiguos de 10 días.

UTILIZACIÓN DEL CLUSTER

El gestor del cluster es Sun Grid Engine/Open Grid Engine. Es el encargado de la distribución de carga entre los nodos.

Realiza las siguientes funciones:

- **Planificación** (scheduling): permite la ejecución de cualquier carga de trabajo cuando los recursos necesarios se encuentren disponibles.
- **Balanceo de carga** (load balancing): distribuye automáticamente la carga entre los nodos de manera que se logre el mejor rendimiento y la utilización óptima de los recursos.
- **Monitorización y contabilidad** (monitoring and accounting): permite la obtención de información acerca de los procesos en ejecución, en lista de espera, resultados, rendimiento, etc.

INICIAR SESIÓN EN EL CLUSTER

Para hacer logon en el cluster es necesario disponer previamente de una cuenta. Para solicitar una cuenta, escribir un correo electrónico a la cuenta hpc@icmat.es indicando los siguientes datos:

- Nombre y apellidos
- Organización a la que se pertenece
- Correo electrónico / teléfono de contacto
- Nombre y apellidos del Investigador principal o responsable en el Instituto
- Descripción del uso y necesidades (librerías, aplicaciones, capacidad de disco y CPU aproximada)

Una vez que se dispone de cuenta, se puede acceder por SSH al cluster:

```
ssh <cuenta>@lovelace.icmat.es
```

INICIAR SESIONES INTERACTIVAS

Es altamente recomendable, para sesiones interactivas, el solicitar al cluster una sesión interactiva en un nodo con baja ocupación mediante el comando "qrsh"

```
[cuenta@master]$ qrsh [-q <cola>]
[cuenta@node02]$
```

De esta manera, obtendremos los beneficios de ejecutar los comandos de sesión en un nodo con baja carga y con recursos suficientes para ello. El nodo maestro se encuentra exclusivamente para la remisión de trabajos por lotes.

Advertencia: el lanzamiento de sesiones interactivas pesadas que contengan trabajos pesados en el nodo maestro puede perjudicar al resto de usuarios, así como presentar malos resultados por saturación del nodo. Además, las CPUs del nodo maestro son más lentas que las CPUs de los nodos de cómputo.

Los nodos de cómputo se encuentran ubicados en las colas siguientes:

```
all.q           Contiene los nodos de cálculo 1 a 20, así como los nodos
                especiales 21 y 22 (xeon phi y nvidia tesla), así como los
                nodos con mayor memoria (101,102,103)

phi.q           Se ejecuta en el nodo de cálculo con dos tarjetas xeon phi,
                cada una de ellas con 60 núcleos de cómputo.

gpu.q           Se ejecuta en el nodo 22, que contiene dos tarjetas GPU
                Nvidia tesla K20m.

bigmem.q        Se ejecuta únicamente en los nodos 101 a 103, que cuentan
                cada uno con 512 Gb de memoria.
```

ENVÍO DE TRABAJOS EN LOTES (BATCH):

Un trabajo (job) representa una tarea a realizar en uno o varios de los nodos del cluster, y contiene la línea de comandos que inicia la tarea. Un trabajo puede especificar requisitos sobre los recursos pero en principio debe ser indiferente sobre el nodo del cluster en que se ejecutará, en tanto en que se cumplan los requisitos de recursos.

```
[cuenta@master]$ qsub -V -b y –cwd <comando o script a ejecutar>
```

Nota: la remisión de trabajos debe hacerse desde el nodo maestro [master].

Siendo los parámetros más típicos los siguientes:

-V	Ejecución manteniendo las variables de entorno de la Shell actual (recomendado)
-b [y|n]	Ejecución especificando si el comando es un binario o un único script (y), que no necesita ser accesible por el host que remite el trabajo, o por el contrario es un script (n) que requiere de ser accedido e interpretado por el equipo que lanza el trabajo.
-cwd	Ejecución tomando el directorio actual como directorio de trabajo del script
-a YYYYMMhhmmss	El trabajo sólo se ejecuta a partir de la hora especificada
-binding	especificación de la afinidad
-i [[host]:]fichero	Redefine la entrada del trabajo al fichero especificado
-e [[host]:]fichero	Redefine la salida de error del trabajo a la especificada
-o [[host]:]fichero	Redefine la salida del trabajo al fichero especificado
-l resource=value	Especifica un recurso necesario para el trabajo, de tal manera que el planificador sólo lanzará el mismo en un nodo que disponga del recurso libre. Pueden especificarse varios requisitos. La página de manual de “complex” muestra cómo obtener la lista de recursos posibles.
-M cuenta@host	Especifica la cuenta de correo o cuentas a las que el servidor enviará un correo si se dan las circunstancias sobre los sucesos al trabajo.
-m [b|e|a|s|n,…]	Define en qué circunstancias notificará a la cuenta de correo especificada acerca de las circunstancias del trabajo (b: al inicio, e: al finalizar, a: al abortarse o re-planificarse, s: al suspenderse, n: en ninguna circunstancia).

-N <nombre>	Establece el nombre del trabajo, que por defecto tomará el nombre del script o del comando ejecutado.
-r [y|n]	Indica si el trabajo debe ejecutarse de nuevo si la ejecución no llega a completarse correctamente (por ejemplo por una caída del nodo).
-verify	En vez de ordenar la ejecución del trabajo, muestra la información acerca de su planificación, como haría qstat –j. Es útil para verificar que todo es correcto antes de lanzar una trabajo de importancia.
-V	Exporta las variables de entorno del Shell actual al trabajo remitido.

NOTA: referirse a la página del manual de qsub para conocer más:

```
[cuenta@master]$ man qsub
```

MONITORIZACIÓN DE TRABAJOS

Los trabajos pueden monitorizarse con el comando “qstat”:

```
[cuenta@master]$ qstat
```

Sin parámetros, muestra toda la cola de trabajos

Las opciones más frecuentes son:

-j [joblist]	Muestra información detallada de los trabajos especificados en la lista, bien mediante los ID de trabajos, nombre o expresiones con comodines. Se muestra la razón del fallo para trabajos en estado “E” (error). Para los trabajos en ejecución se muestra el consumo de recursos actual.
-f	Muestra parámetros de todas las colas individuales. Se suele utilizar con el parámetro –ne (not empty) para que no muestre las colas sin trabajo.
-r	Muestra información acerca de los recursos utilizados por los trabajos mostrados.

```
-s {p|r|s|z|hu    Filtra por estados:
|ho|hs|hd|hj|ha   p –
|h|a}[+]          r – en ejecución (running)
                  s – planificado (scheduled)
                  z – finalizado recientemente
                  hu|ho|hs|hd|hj|ha – retenido (on hold)

-explain a|A|c|E  Solicita información adicional acerca de los trabajos en
                  estado especiales:
                  a – alarma
                  A – suspendidos
                  c – configuración ambigüa
                  E – error

-t                Muestra información acerca de las sub-tareas de un trabajo
                  en paralelo.

-u <usuario>      Muestra información acerca de los trabajos de una cuenta de
                  usuario.
```

NOTA: más información acerca de estos comandos en la página del manual, y en la guía de usuario de Sun Grid Engine.

CANCELACIÓN DE TRABAJOS

La cancelación de trabajos se realiza mediante el comando qdel:

```
[cuenta@master]$ qdel [lista de trabajos]
```

Los trabajos pueden especificarse por identificador, nombre o mediante caracteres comodín.

Las opciones más utilizadas:

```
-f                Fuerza la eliminación del trabajo incluso si el nodo donde
                  se está ejecutando no responde.

-u <usuario>      Elimina todos los trabajos del usuario especificado.

-t <tareas>       Elimina las tareas especificadas de un trabajo múltiple.
```

MODIFICACIÓN DE LOS PARÁMETROS DE LOS TRABAJOS

Se pueden modificar las características de los trabajos enviados a una cola mediante el comando qalter:

```
[cuenta@master]$ qalter [opciones] <lista de trabajos> [argumentos]
```

También pueden modificarse todos los trabajos de una cuenta o cuentas de usuario:

```
[cuenta@master]$ qalter [opciones] –u <lista de usuarios> | -u all [argumentos]
```

Las modificaciones más frecuentes corresponden con los siguientes argumentos:

-a YYYYMMhhmmss	Redefine la hora a la que se debe ejecutar el trabajo.
-cwd	Redefine el directorio de trabajo
-i [[host]:]fichero	Redefine la entrada del trabajo al fichero especificado
-e [[host]:]fichero	Redefine la salida de error del trabajo a la especificada
-o [[host]:]fichero	Redefine la salida del trabajo al fichero especificado
-l resource=value	Especifica un recurso necesario para el trabajo, de tal manera que el planificador sólo lanzará el mismo en un nodo que disponga del recurso libre. Pueden especificarse varios requisitos.
-h u	Establece el trabajo en el estado “on hold” de manera que no se ejecutará, hasta que se libere con el comando qalter –h U o el comando qrls. Otro medio alternativo es utilizar el comando qhold.
-M cuenta@host	Especifica la cuenta de correo o cuentas a las que el servidor enviará un correo si se dan las circunstancias sobre los sucesos al trabajo.
-m [b|e|a|s|n,…]	Define en qué circunstancias notificará a la cuenta de correo especificada acerca de las circunstancias del trabajo (b: al inicio, e: al finalizar, a: al abortarse o re-planificarse, s: al suspenderse, n: en ninguna circunstancia).
-N <nombre>	Establece el nombre del trabajo, que por defecto tomará el nombre del script o del comando ejecutado.

```
-r [y|n]          Modifica si el trabajo debe ejecutarse de nuevo si la
                  ejecución no llega a completarse correctamente (por ejemplo
                  por una caída del nodo).
```

OBTENCIÓN DE LOS RESULTADOS DE LOS TRABAJOS

Para los trabajos en BATCH, se generará en el directorio especificado como directorio de ejecución un fichero con la salida estándar del trabajo, que se llamará como el nombre del trabajo y acabado en .o<ID del trabajo>:

```
[<cuenta>@master ~]$ echo "hostname" | qsub -q all.q -N prueba
Your job 864 ("prueba") has been submitted
[<cuenta>@master ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    864 0.00000 prueba     acaso        qw    01/19/2017 17:34:10                                 1
[<cuenta>@master ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    864 0.55500 prueba     acaso        r     01/19/2017 17:34:18 all.q@node14.icmat              1
[<cuenta>@master ~]$  ls prueba*
prueba.e864  prueba.o864
```

También se generará un fichero equivalente para la salida de error acabado en .e<ID>, a menos que hayamos especificado el parámetro –j, en cuyo caso se generará un único fichero (.o).

CREACIÓN DE SCRIPTS PARA EJECUCIÓN DE TRABAJOS

Los scripts de creación de trabajos permiten una serie de características especiales, mediante comentarios, que son interpretados por qsub para evitar tener que recordar los diversos elementos y opciones de planificación.

La sintaxis típica de un script es como sigue:

```
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#$ -V
<Comando o comandos a ejecutar, interpretados por la shell>
```

La primera línea especifica el intérprete de comandos (Shell) a utilizar en la ejecución del propio script.

Las líneas que comienzan con #$ indican a qsub que lo que sigue a continuación es una opción de qsub, con el mismo significado que en comando.

GESTIÓN DE VARIABLES DE ENTORNO PARA TRABAJAR CON DISTINTAS APLICACIONES Y LIBRERÍAS

Para facilitar la gestión de los entornos y posibilitar el uso de diferentes combinaciones de compiladores, librerías y aplicaciones, se ha optado por el uso de "environment-modules".

Este componente permite cargar y descargar las distintas configuraciones posibles con facilidad. Se basa en el comando "module" para realizar esta gestión.

Las operaciones básicas son: ver los módulos disponibles, listar los módulos cargados, cargar un módulo o descargarlo.

Mostrar los módulos disponibles:

```
[cuenta@node20]$ module avail
------------------------------ /etc/modulefiles ------------------------------
openmpi-1.8-x86_64 openmpi-x86_64

---------------------------- /LUSTRE/apps/Modules/ ----------------------------
compilers/gcc447 matlab/2010b     matlab/2015a     mpich-x86_64
```

Para cargar un módulo basta con ejecutar:

```
[cuenta@master]$ module load <módulo>
```

Para verificar los módulos que tenemos cargados, utilizaremos el comando "list"

```
[cuenta@master]$ module list
```

Por último, para evitar conflictos entre módulos, podemos utilizar el comando "unload"

```
[cuenta@master]$ module unload <módulo>
```

Este comando se puede incluir también en scripts para cargar los módulos correspondientes.

REPORTAR SOLICITUDES, INCIDENCIAS Y PROBLEMAS

Dirigirse a la cuenta por correo electrónico a:

hpc@icmat.es

MÁS INFORMACIÓN

Manuales

Sitio del proyecto Open Source http://gridscheduler.sourceforge.net/

Información en Wikipedia: https://es.wikipedia.org/wiki/Sun_Grid_Engine